0083_unpack_list_tuple

May 8, 2018

1 Automatically passing unpacked lists or tuples to a function (or why do you see * before lists and tuples)

Imagine we have a very simple function that adds three numebrs together:

```
In [1]: def add_three_numbers (a, b, c):
            return a + b + c
```

We would normally pass separate numebrs to the function, e.g.

```
In [2]: add_three_numbers (10, 20, 35)

Out[2]: 65
```

But what if our numbers are in a list or a tuple. If we pass that as a list then we are passing only a single argument, and the function declares an error:

```
In [3]: my_list = [10, 20, 35]

        add_three_numbers (my_list)


        ---------------------------------------------------------------------------

        TypeError                                 Traceback (most recent call last)

        <ipython-input-3-900947bff326> in <module>()
          1 my_list = [10, 20, 35]
          2
    ----> 3 add_three_numbers (my_list)


        TypeError: add_three_numbers() missing 2 required positional arguments: 'b' and 'c'
```

Python allows us to pass the list or tuple with the instruction to unpack it for input into the function. We instruct Python to unpack the list/tuple with an asterix before the list/tuple:

```
In [4]: add_three_numbers (*my_list)

Out[4]: 65
```